

V1P2

2. Pentru definiția de mai jos a subprogramului `f`, ce se afișează ca urmare a apelului `f(121,1);?` (6p.)

<pre>//C void f(long n, int i) { if(n!=0) if(n%3>0) { printf("%d",i); f(n/3,i+1); } }</pre>	<pre>//C++ void f(long n, int i) { if(n!=0) if(n%3>0) { cout<<i; f(n/3,i+1); } }</pre>
--	---

V2

2. Pentru definiția de mai jos a subprogramului `f`, ce se afișează ca urmare a apelului `f(12345);?` (6p.)

<pre>//C void f(long n) { printf("%d",n%10); if(n!=0) { f(n/100); printf("%d",n%10); } }</pre>	<pre>//C++ void f(long n) { cout<<n%10; if(n!=0) { f(n/100); cout<<n%10; } }</pre>
--	--

V3

2. Pentru definiția alăturată a subprogramului `f`, ce se afișează ca urmare a apelului `f(26);?` (6p.)

```
void f (int x)
{
  if(x>0)
  if(x%4==0)
    { cout<<'x'; | printf("%c",'x');
      f(x-1); }
  else
    { f(x/3);
      cout<<'y'; | printf("%c",'y');
    }
}
```

V4

2. Pentru definiția alăturată a subprogramului `f`, ce se afișează ca urmare a apelului `f(15,2);?` (6p.)

```
void f (int n, int x)
{ if(x>n)
  cout<<0; | printf("%d",0);
  else
  if(x%4<=1) f(n,x+1);
  else
  { f(n,x+3);
    cout<<1; | printf("%d",1);
  }
}
```

V5

2. Pentru subprogramul `f` definit mai jos, ce se afișează ca urmare a apelului `f(3,17)?` (6p.)

<pre>//C void f (int a, int b) { if(a<=b) {f(a+1,b-2); printf("%c",'*');} else printf("%d",b); }</pre>	<pre>//C++ void f (int a, int b) { if(a<=b) { f(a+1,b-2); cout<<'*';} else cout<<b; }</pre>
--	---

V6

2. Se consideră subprogramul alăturat:
Ce valoare are $f(128, 2)$?

(6p.)

```
int f(int a, int b){
    if (b<1) return -1;
    else
        if (a%b==0)
            return 1+f(a/b,b);
        else
            return 0; }
```

V7

1. Se consideră subprogramul f definit alăturat. Ce se va afișa în urma apelului $f(12345);?$ (4p.)

```
void f(long int n)
{ if (n!=0)
  {if (n%2 == 0)
    cout<<n%10; | printf("%d",n%10);
    f(n/10);
  }
}
```

a. 513

b. 24

c. 42

d. 315

V8

2. Se consideră subprogramul f , descris alăturat. Ce se va afișa în urma apelului $f(3);?$ (6p.)

```
void f(int n)
{ if (n!=0)
  { if (n%2==0)
    cout<<n<<' '; | printf("%d ",n);
    f(n-1);
    cout<<n<<' '; | printf("%d ",n);
  }
  else cout<<endl; | printf("\n");
}
```

V9

1. Se consideră subprogramul f cu definiția alăturată. Ce valoare are $f(1213111, 1)$?

(4p.)

```
int f (long int n, int k){
    if (n!=0)
        if (n%10==k)
            return 1+f(n/10,k);
        else return 0;
    else return 0;}
```

a. 5

b. 3

c. 2

d. 1

V10

1. Se consideră subprogramul cu definiția alăturată. Ce valoare are $f(3, 1)$?

(4p.)

```
int f(int n,int y)
{ if (n!=0)
  { y=y+1;
    return y+f(n-1,y);
  }
  else return 0;
}
```

a. 9

b. 6

c. 7

d. 8

V11

2. Care este valoarea expresiei $f(23159)$ pentru funcția f , cu definiția alăturată? (6p.)
- ```

int f(int n){
 int c;
 if (n==0) return 9;
 else
 {c=f(n/10);
 if (n%10<c) return n%10;
 else return c;
 }
}

```

V12

1. Se consideră subprogramul  $P$ , definit alăturat. Știind că valoarea variabilei întregi  $a$  este înainte de apel 4, care este valoarea ei după revenirea din apelul  $P(a)$ ? (4p.)
- ```

void P(int &x)
{ x=x+5; }

```
- a. 10 b. 4 c. 9 d. 5

V13

1. Fie subprogramul fct definit alăturat, parțial. Inițial, variabile întregi a, b și c au valorile $a=8, b=31$ și $c=9$, iar după apelul $fct(a,b,c)$, valorile celor trei variabile sunt $a=9, b=31$ și $c=40$. Care poate fi antetul subprogramului fct ? (4p.)
- ```

void fct(....)
{ x++; y--;
 z=x+y;
}

```
- a. `void fct(int &x,int &y,int &z)`                      b. `void fct(int x,int &y,int &z)`  
c. `void fct(int x,int y,int z)`                      d. `void fct(int &x,int y,int &z)`

V14

1. Care dintre următoarele variante reprezintă antetul corect al unui subprogram care primește prin parametri  $x$  și  $y$  două numere întregi și furnizează prin parametrul  $m$  cea mai mică dintre cele două valori  $x$  și  $y$ ? (4p.)
- a. `int minim(int x,int y,int m)`                      b. `void minim(int x,int y,int &m)`  
c. `int minim(int x,int y)`                      d. `void minim(int x,int y,int m)`

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Pentru funcțiile  $f1$  și  $f2$  definite alăturat, stabiliți care este valoarea lui  $f2(41382)$ ? (6p.)
- ```

long f1(int c)
{ if (c%2==1) return 1;
  else return 2;
}

long f2(long n)
{ if (n==0) return 0;
  else return f1(n%10)+f2(n/10);
}

```

V15

2. Pentru funcțiile f și g definite mai jos, stabiliți care este rezultatul returnat la apelul $f(6)$? (6p.)

```

long g(long x)
{ if (x>9)
    return (x/10 + x%10);
  else
    return x
}

long f(int c)
{ if (c<1)
    return 1;
  else
    return g(c+f(c-1));
}

```

V16

1. Funcția F are definiția alăturată. Ce valoare are $F(3)$? (4p.)
- ```
int F(int n)
{if(n==0 || n==1) return 1;
 else
 return 2*F(n-1)+2*F(n-2);}
```
- a. 1                      b. 12                      c. 6                      d. 10

V17

1. Subprogramul  $f$  are definiția alăturată. Ce se va afișa în urma apelului  $f(12345)$ ? (4p.)
- ```
void f(long n)
{if (n>9)
 {cout<<n/100; | printf("%d",n/100);
  f(n/10);
 }
}
```
- a. 1231210 b. 123121 c. 1234123121 d. 123

V18

1. Funcția f are definiția alăturată. Ce se va afișa în urma apelului $f(12345,0)$? (4p.)
- ```
void f(long n, int i)
{if (i<n%10)
 {cout<<n%10; | printf("%d",n%10);
 f(n/10,i+1);
 }
}
```
- a. 54321                      b. 543                      c. 54                      d. 5432

V19

1. Funcția  $F$  are definiția alăturată. Ce valoare are  $F(18)$ ? (4p.)
- ```
int F(int x){
  if (x<=1) return x;
  else
    return x+F(x-2);}
```
- a. 90 b. 171 c. 1 d. 18

V20

1. Funcția F are definiția alăturată. Ce valoare are $F(5)$? (4p.)
- ```
int F(int x)
{if(x!=0) return x+F(x-1);
 else
 return x;
}
```
- a. 5                      b. 10                      c. 15                      d. 6

V21

2. Se consideră subprogramul,  $f$ , definit alăturat. Ce valoare are  $f(100)$ ? (6p.)
- ```
int f(int n)
{ if(n==0) return 0;
  else return n%2+f(n/2);
}
```

V22

2. Considerăm subprogramul recursiv definit alăturat. Ce se va afișa în urma apelului: $f('C')$ (4p.)

```
void f(char c)
{ if (c>'A') f(c-1);
  cout<<c; | printf("%c",c);
  if (c>'A') f(c-1);
}
```

V23

2. Pentru subprogramul `suma` definit alăturat, scrieți valoarea expresiei `suma(5,4)`. (4p.)

```
int suma (int a,int b)
{ if (a==0 && b==0) return 0;
  else if (a==0) return 1+suma(a,b-1);
  else return 1+suma(a-1,b);
}
```

V24

2. Funcția f are definiția alăturată.

a) Ce valoare are $f(17)$?

(3p.)

b) Ce valoare are $f(22)$?

(3p.)

```
int f(int n)
{if (n<=9) return 0;
 if (n%4==0) return 0;
 return 1+f(n-3);
}
```

V25

2. Funcția f are definiția alăturată:

a) Ce valoare are $f(16)$?

(3p.)

b) Scrieți cea mai mare valoare de două cifre pe care o poate avea n astfel încât $f(n)$ să fie egal cu 2. (3p.)

```
int f(int n)
{ if (n<=0) return -1;
  if (n%2==0) return 0;
  if (n%3==0) return 0;
  return 1+f(n-10);
}
```

V26

2. Subprogramul `afis` este definit alăturat. Ce se afișează ca urmare a apelului `afis(8)`;

(4p.)

```
void afis (int n)
{ cout<<n; | printf("%d",n);
  for (int i=n/2;i>=1;i--)
    if (n%i==0) afis(i);
}
```

V27

2. Subprogramul `scrie` este definit alăturat.

Ce se afișează ca urmare a apelului `scrie(1,7)`;

(6p.)

```
void scrie (int x,int y)
{cout<<x<<y; | printf("%d%d",x,y);
  if (x<y)
  {scrie(x+1,y-1);
   cout<<(x+y)/2; | printf("%d", (x+y)/2);
  }
}
```

V28

2. Subprogramul `f` este definit alăturat.
Ce valoare are `f(8,4)`? (4p.)

```
int f (int x,int y)
{ if(x<y) return 1+f(x+1,y);
  if(y<x) return 1+f(y+1,x);
  return 1; }
```

V29

2. Ce se afișează ca urmare a
apelului `p(123)`; dacă
subprogramul `p` are definiția
alăturată. (6p.)

```
void p (int x)
{cout<<x; | printf("%d",x);
 if(x!=0){p(x/10);
 cout<<x%10; | printf("%d",x%10);}}
```

V30

2. Subprogramul `f` este definit alăturat.
Ce se afișează ca urmare a apelului
`f(1,4)`; (6p.)

```
void f (int x,int y)
{ for (int i=x;i<=y;i++)
  {cout<<i;f(i+1,y);}
}
```

V31

2. Se consideră subprogramul recursiv `f1`
definit alăturat. Ce se va afișa în urma
apelului `f1(5)`? (6p.)

```
void f1(int x)
{ if (x<=9)
  { cout<<x+1; | printf("%d",x+1)
    f1(x+2);
    cout<<x+3; | printf("%d",x+3)
  }
}
```

V32

2. Subprogramul `afis` este definit alăturat.
Ce se va afișa în urma apelului
`afis(17)`? (6p.)

```
void afis(int x)
{ if (x>3)
  { cout<<x-1; | printf("%d",x-1);
    afis(x/3);
    cout<<x+1; | printf("%d",x+1);
  }
}
```

V33

2. Subprogramul `re` este definit alăturat. Ce valoare
are `re(1)`? Dar `re(14)`? (6p.)

```
int re(int i)
{
  if (i<9) return 3+re(i+2);
  else
    if (i==9) return -2;
    else return 1+re(i-1);
}
```

V34

2. Se consideră definit subprogramul f .

Scrieți două valori naturale, x_1 și x_2 , ($x_1 < 12$ și $x_2 < 12$) pentru care $f(x_1) = f(x_2)$.

(6p.)

```
int f(int i)
{
    if (i > 12) return 1;
    else return 1 + f(i + 2);
}
```

V35

2. Subprogramul f este definit alăturat. Ce se va afișa în urma executării secvenței de mai jos?

$a = 3, b = 9; f(a, b); f(b, b);$

(6p.)

```
void f(int &a, int b)
{
    a = a - 5; b = a - 2;
    cout << a << b; | printf("%d%d", a, b);
}
```

V36

1. Subprogramul $scif$ returnează suma cifrelor unui număr natural transmis ca parametru.

Care este valoarea expresiei $scif(scif(518) + scif(518))$?

(4p.)

- a. 14 b. 10 c. 28 d. 1

V37

2. Considerăm subprogramul f definit alăturat. Ce valoare are $f(11, 7)$?

(6p.)

```
int f(int x, int y)
{
    if (x <= y) return x - y;
    return f(y - x, x - 1) + 3;
}
```

V38

2. Pentru definiția alăturată a subprogramului sc , stabiliți ce valoare are $sc(901324)$?

(6p.)

```
int sc(long x)
{
    if (x < 10) return x;
    return sc(x / 10) + x % 10;
}
```

V39

2. Pentru definiția alăturată a subprogramului f , ce valoare are $f(8)$?

(4p.)

```
int f(int x)
{
    if (x <= 4) return x * x - 3;
    return f(x - 3) + 4;
}
```

V40

2. Pentru definiția alăturată a subprogramului f , stabiliți ce valoare are $f(23461)$?

(4p.)

```
int f(int x)
{
    if (x < 10) {
        if (x % 2 != 0) return 0;
        return x;
    }
    if ((x % 2 != 0))
        return f(x / 10);
    return f(x / 10) + x % 10;
}
```

V41

2. Subprogramul f are definiția alăturată. Ce valoare are $f(7, 2)$? Dar $f(35, 2)$?

(6p.)

```
int f(int x, int y)
{
    if (x % y == 0) return y;
    else return f(x, y + 1);
}
```

V42

2. Subprogramul f are definiția alăturată. Ce valoare are $f(7)$? Dar $f(100)$? (6p.)
- ```
int f(int x)
{if(x%6==0)return x;
 else return f(x-1);
}
```

V43

2. Subprogramul  $f$  are definiția alăturată. Ce valoare are  $f(3)$ ? Dar  $f(10)$ ? (6p.)
- ```
int f(int x)
{if(x==0)return 0;
  else return f(x-1)+2;
}
```

V44

2. Subprogramul f are definiția alăturată. Ce valoare are $f(5,10)$? (6p.)
- ```
int f(int x,int y)
{if(x==y)return x;
 else if(x<y)return f(x+1,y-1);
 else return f(x-1,y);
}
```

V45

2. Subprogramul  $f$  are definiția alăturată. Ce valoare are  $f(4)$ ? Dar  $f(11)$ ? (6p.)
- ```
int f(int x)
{if(x<1)return 1;
  else return f(x-3)+1;
}
```

V46

2. Funcția f are definiția alăturată. Dacă $f(x)$ are valoarea 10100, care este valoarea lui x ? (6p.)
- ```
long f(int n)
{
 if(n<=0) return 0;
 else return f(n-1)+2*n;
}
```

V47

1. Valorile memorate de componentele vectorului  $v$ , cu indicii de la 0 la 5, sunt, în această ordine: 183, 212, 453, 18, 42, 83. Care dintre următoarele apeluri ale subprogramului  $tab$  cu definiția alăturată, **NU returnează** un număr par? (4p.)
- ```
int tab (int i,int v[], int c)
{ int n;
  if(i<0) return 0;
  else
  { n=v[i];
    while(n!=0)
    { if(n%10==c%10)
      return 1+tab(i-1,v,c%10);
      n=n/10;
    }
    return tab(i-1,v,c%10);
  }
}
```
- a. $tab(4,v,218)$ b. $tab(5,v,72)$ c. $tab(4,v,123)$ d. $tab(5,v,71)$

V48

1. Valorile memorate de componentele vectorului **v**, cu indicii de la 0 la 5, sunt, în această ordine: 183, 212, 453, 18, 42, 83. Care este apelul corect al subprogramului **tablou** alăturat pentru ca, în urma executării apelului, valoarea variabilei întregi **x** să fie 2?

(4p.)

- a. **x=tablou(2,v,2);**
c. **x=tablou(5,v,2);**

```
int tablou (int i,int v[], int c)
{int n;
 if(i<0) return 0;
 else
 { n=v[i];
  while(n>9)
  { if(n%10==c)
    return 1+tablou(i-1,v,c);
    n=n/10;
  }
  return tablou(i-1,v,c);
 }
}
```

b. **x=tablou(2,v,5);**
d. **x=tablou(2,v,8);**

V49

1. Valorile memorate de componentele vectorului **v**, cu indicii de la 0 la 5, sunt, în această ordine: 420, 48, 635, 628, 837, 93. Care este apelul corect al subprogramului **tablou** alăturat pentru ca, în urma executării apelului, valoarea variabilei întregi **x** să fie 3?

(4p.)

- a. **x=tablou(3,v);**
c. **x=tablou(5,v);**

```
int tablou (int i,int v[])
{
 if(i<0) return 0;
 else {
  int n=v[i];
  while(n) {
   if(n%2) return 1+tablou(i-1,v);
   n=n/10; }
  return tablou(i-1,v); }
}
```

b. **x=tablou(4,v);**
d. **x=tablou(0,v);**

V50

1. Valorile memorate de componentele vectorului **v**, cu indicii de la 0 la 5, sunt, în această ordine: 973, 51, 871, 350, 691, 15. Care este apelul corect al subprogramului **tablou** alăturat pentru ca, în urma executării apelului, valoarea variabilei întregi **x** să fie 3?

(4p.)

- a. **x=tablou(4,v,5);**
c. **x=tablou(5,v,1);**

```
int tablou (int i,int v[], int c)
{
 if(i<0) return 0;
 else {
  int n=v[i];
  while(n) {
   if(n%10==c)
    return 1+tablou(i-1,v,c);
   n=n/10; }
  return tablou(i-1,v,c); }
}
```

b. **x=tablou(5,v,5);**
d. **x=tablou(0,v,4);**

V51

1. Care este antetul corect al subprogramului **cifre** care primește prin intermediul primului parametru **x** un număr natural și furnizează prin intermediul celui de-al doilea parametru, **y**, suma cifrelor numărului natural **x**?

(4p.)

- a. **void cifre(int x, int &y)**
c. **void cifre(int x, int y)**
- b. **int cifre(int x)**
d. **int cifre(int &x, int y)**

V52

2. Se consideră subprogramul `f`, definit alăturat. Ce valoare are `f(4)`? Dar `f(100)`? (6p.)
- ```
long f(unsigned int n)
{ if (n==0) return 0;
 else return n+f(n-1);
}
```

V53

1. Subprogramul `f` realizează interschimbarea valorilor a 2 variabile întregi. Care este antetul corect al subprogramului `f`? (4p.)
- a. `void f(int &x, int &y)`                      b. `int f(int x,int y)`  
 c. `void f(int x, int &y)`                      d. `void f(int &x, int y)`

V53

2. Se consideră subprogramul `f`, definit alăturat. Câte valori ale lui `n` aparținând intervalului `[10, 20]` au proprietatea că la apelul `f(n)` se obține valoarea 0? (6p.)
- ```
int f(unsigned int n)
{if (n==0) return 0;
 else if (n%2==0) return n%10+f(n/10);
 else return f(n/10);
}
```

V54

2. Se consideră subprogramul `f` definit alăturat. Ce se afișează la apelul `f(1)`? (6p.)
- ```
void f(int i)
{if (i<=5) {
 cout<<i<<" " ;| printf("%d ",i);
 f(i+1);
 cout<<i/2<<" " ;| printf("%d ",i/2);
 }
}
```

V55

- Se consideră subprogramul `f`, definit alăturat. Câte valori ale parametrului `n` aparținând intervalului `[1,100]` au proprietatea că la apelul `f(n)` se obține o valoare nenulă? (6p.)
- ```
int f(unsigned int n)
{ if (n>20) return 0;
  else return 5+f(n+5);
}
```

V56

2. Se consideră subprogramul `f`, definit alăturat. Ce valoare are `f(21114,1)`? (6p.)
- ```
int f(int n,int c)
{
 if (n==0) return 0;
 if (n%10==c) return n%100+f(n/10,c);
 return f(n/10,c);
}
```

V57

2. Se consideră subprogramul `f`, definit alăturat. Ce se va afișa la apelul `f(38)`? (6p.)
- ```
void f(int x){
  if(x){
    if(x%3==0){
      cout<<3; | printf("3");
      f(x/3);
    }
    else{
      f(x/3);
      cout<<x%3; | printf("%d",x%3);
    }
  }
}
```

V58

2. Se consideră subprogramul f , definit alăturat. Ce valoare are $f(3713, 3)$?
(6p.)

```
int f(int n, int c) {
    if (n == 0)
        return 0;
    if (n % 10 == c)
        return f(n / 10, c) * 10 + c;
    return f(n / 10, c);
}
```

V59

2. Se consideră subprogramul f , definit alăturat. Ce valoare are $f(261, 31)$?
(6p.)

```
int f(int a, int b)
{
    if (a < 10)
        return b;
    return f(a / 10, b) * 10 + b + 1;
}
```

V60

2. Se consideră subprogramul f , definit alăturat. Ce valoare are $f(398, 2008)$?
(6p.)

```
int f(int a, int b) {
    if (2 * a >= b)
        return 0;
    if (b % a == 0)
        return b - a;
    return f(a + 1, b - 1);
}
```

V61

2. Se consideră tabloul unidimensional a definit global, ce memorează elementele $a_1=12$, $a_2=35$, $a_3=2$, $a_4=8$ și subprogramul f , definit alăturat. Ce valoare are $f(4)$?
(6p.)

```
int f(int x)
{ if (x >= 1)
    return (a[x] + f(x - 1)) % 10;
  else
    return 0; }
```

V62

2. Considerăm subprogramul f , definit alăturat. Care va fi valoarea variabilei globale x după apelul $f(4962, x)$, dacă înainte de apel, x are valoarea 0?
(6p.)

```
void f1(int n, int &a)
{ int c;
  if (n != 0) { c = n % 10;
               if (a < c) a = c;
               f1(n / 10, a); } }
```

V63

2. Se consideră tabloul unidimensional a definit global, ce memorează elementele $a_1=1$, $a_2=2$, $a_3=0$ și subprogramul f , definit alăturat. Ce valoare are $f(3, 3)$?
(6p.)

```
int f(int b, int i)
{ if (i >= 1)
    return f(b, i - 1) * b + a[i];
  else return 0; }
```

V64

2. Se consideră subprogramul **f**, definit alăturat. Ce se va afișa în urma apelului **f(5,0)**?

(6p.)

```
void f(int i,int j)
{ if(j<=9) f(i,j+1);
  printf("%d*d=%d\n",i,j,i*j);/
  cout<<i<<' '*<<j<<'='<<i*j<<endl;
}
```

V65

2. Se consideră funcția **f**, definită alăturat. Ce se afișează ca urmare a executării secvenței de mai jos ?
a=4; b=18;
printf("%d",f(a,b));/ cout<<f(a,b);
printf("%d %d",a,b);/ cout<<a<<b;

(6p.)

```
int f(unsigned int &a,
unsigned int &b)
{ while (a !=b)
  if (a>b) a=a-b;
  else b=b-a;
  return a;}
```

V66

2. Se consideră subprogramul recursiv definit alăturat. Ce valoare are **functie(4)**?

(6p.)

```
int functie(int x)
{
  if (x<=0) return 3 ;
  else return functie(x-3)*4 ;}
```

V67

2. Se consideră subprogramul recursiv definit alăturat. Pentru ce valoare a parametrului **u**, **alfa(u)** are valoarea 25 ? Pentru **u=6**, ce valoare returnează **alfa(u)**?

(6p.)

```
int alfa(int u)
{if (u==0) return 3;
 else return alfa(u-1)+3*u-2;
}
```

V68

2. Se consideră subprogramul recursiv definit alăturat. Ce valori vor fi afișate pe ecran în urma apelului **gama(6)**?

(6p.)

```
void gama(int n)
{int i;
  if(n>=3)
  {for(i=3;i<=n;i++)
    printf("%d ",n); | cout<<n<<" ";
    gama(n-3);}}
```

V69

2. Se consideră subprogramul recursiv definit alăturat. Ce valori vor fi afișate pe ecran în urma apelului **beta(15)**?

(6p.)

```
void beta(int n)
{if (n!=1)
  {printf("%d ",n); | cout<<n<<" ";
   if (n%3==0) beta(n/3);
   else beta(2*n-1);}
  else printf("%d",1); | cout<<1;}
```

V70

2. Se consideră subprogramul recursiv definit alăturat.
Care este cea mai mică valoare de 2 cifre a lui u pentru care
funcția $\text{bac}(u, 2)$ are valoarea 1 ? (6p.)

```
int bac(int u, int x)
{if (u<x) return 0;
 if (x==u) return 1;
 else
  if (u%x==0) return 0;
  else
   return bac(u,x+1);}
```

V71

2. Ce valoare va avea variabila întreagă x , în
urma apelului $F(1, x)$, știind că, înainte de
apel, variabila x are valoarea 0, iar
subprogramul F este definit alăturat?

(6p.)

```
void F(int i, int &x)
{ if (i <= 10)
  {   if(i % 2)  x = x + 2;
      else      x = x - 1;
      F(i + 1, x);
  }
}
```

V72

2. Scrieți ce se va afișa în urma
executării subprogramului alăturat,
la apelul $F(57)$. (6p.)

```
void F(int x)
{ if(x != 0)
  { F(x/2);
    cout << x%2; | printf("%d", x%2);
  }
}
```

V73

2. Ce se va afișa în urma executării
subprogramului alăturat, la apelul
 $F(56)$? (6p.)

```
void F(int x)
{ if(x)
  { F(x/2);
    cout << x%10; | printf("%d", x%10);
  }
}
```

V74

2. Se consideră funcția Suma , definită
alăturat. Ce valoare are $\text{Suma}(8)$? Dar
 $\text{Suma}(11)$? (6p.)

```
int Suma(int x)
{ if(x == 1) return 0;
  if(x%2==0) return Suma(x-1)+(x-1)*x;
  return Suma(x-1)-(x-1)*x;
}
```

V75

1. Ce se va afișa în urma apelului
 $F(2758)$, pentru funcția F
definită alăturat? (4p.)

```
int F(int x)
{
  if(x == 0) return 0;
  if(x%10%2 == 0) return 2 + F(x/10);
  return 10 - F(x/10);
}
```

a. 0

b. 20

c. 12

d. 4

V76

2. Pentru definiția alăturată a subprogramului `sub`, stabiliți ce valoare are `sub(132764)`. (6p.)
- ```
long sub(long n)
{
 if (n!=0)
 if (n%2!=0) return n%10*sub(n/10);
 else return sub(n/10);
 else return 1;
}
```

V77

2. Pentru definiția alăturată a subprogramului `sub`, stabiliți ce valoare are `sub(132764)`. (6p.)
- ```
int sub(long n)
{
    if (n!=0)
        if (n%2!=0) return n%10+sub(n/10);
        else return sub(n/10);
    else return 0;
}
```

V78

2. Pentru definiția alăturată a subprogramului `sub`, stabiliți care este valoarea expresiei `sub(123986)`. (6p.)
- ```
int sub(long n)
{
 if (n!=0)
 if (n%2!=0) return 1+sub(n/10);
 else return sub(n/10);
 else return 0;
}
```

V79

2. Pentru definiția alăturată a subprogramului `f`, ce valoare are expresia `f(1209986)`? (6p.)
- ```
int f(long x)
{
    int y,z;
    if (x==0) return x;
    else {y=x%10;
          z=f(x/10);
          if (y>z) return y;
          else return z;
    }
}
```

V80

2. Pentru definiția alăturată a subprogramului `sub`, ce valoare are `sub(132764)`? (6p.)
- ```
int sub(long n)
{
 if (n!=0)
 if (n%2!=0) return n%10+sub(n/10);
 else return sub(n/10);
 else return 0;
}
```

V81

2. Pentru definiția alăturată a subprogramului `f`, scrieți ce valoare are `f(5552,5)`. (6p.)
- ```
int f(int x,int y)
{
    if (x==0) return 0;
    else
        if (x%10==y) return f(x/10,y)+1;
        else return f(x/10,y);
}
```

V82

2. Pentru definiția alăturată a subprogramului f , stabiliți ce valoare are $f(123)$? (6p.)
- ```
int f(int x)
{ if(x==0) return 0;
 else
 if(x%2==0) return 1+f(x/10);
 else return 2+f(x/10);
}
```

V83

2. Pentru definiția alăturată a subprogramului  $f$ , stabiliți ce valoare are  $f(123)$ ? (6p.)
- ```
int f(int x)
{ if(x==0) return 0;
  else
    if(x%2==0) return 3+f(x/10);
    else return 4+f(x/10);
}
```

V84

2. Pentru definiția alăturată a subprogramului f , stabiliți ce valoare are $f(100)$? (6p.)
- ```
int f(int x)
{ if(x==0) return 1;
 else
 return 1+f(x-1);
}
```

V85

2. Pentru definiția alăturată a subprogramului  $f$ , scrieți ce valoare are  $f(100)$ . (6p.)
- ```
int f(int x)
{ if(x==50) return 1;
  else
    return 2+f(x-1);
}
```

V86

2. Pentru definiția alăturată a subprogramului f , scrieți ce valoare are $f(100)$. (6p.)
- ```
int f(int x)
{ if(x==50) return 1;
 else
 return 2+f(x-1);
}
```

V87

2. Se consideră subprogramul  $f$ , definit alăturat. Ce se afișează la apelul  $f(4)$ ? (6p.)
- ```
void f(int n)
{
  if(n>0)
  {
    cout<<n; | printf("%d",n);
    f(n-1);
    cout<<n; | printf("%d",n);
  }
}
```

V88

2. Se consideră subprogramul f , definit alăturat. Ce valoare are $f(4)$?

(6p.)

```
int f(int n)
{
    if (n==0) return 1;
    else if (n==1) return 2;
    else return f(n-1)-f(n-2);
}
```

V89

2. Se consideră subprogramul f , definit alăturat. Ce valoare are $f(4)$?

(6p.)

```
long f(int n)
{
    if (n==0) return 0;
    else return n*n+f(n-1);
}
```

V90

2. Se consideră subprogramul f , definit alăturat. Ce se afișează la apelul $f('a')$?

(6p.)

```
void f(char c)
{
    if (c != 'e')
    {
        f(c+1);
        cout<<c; | printf("%c",c);
    }
}
```

V91

2. Subprogramul recursiv alăturat este definit incomplet. Scrieți expresia care poate înlocui punctele de suspensie astfel încât, în urma apelului, $f(12)$ să se afișeze șirul de valori:

12 6 3 1 1 3 6 13. (6p.)

```
void f(int i)
{ if ( . . . )
{ printf("%d ",i); | cout<<i<<' ';
  f(i/2);
  printf("%d ",i); | cout<<i<<' ';
}
}
```

V92

2. Se consideră definit subprogramul f .
a) Ce se va afișa în urma apelului $f(14)$?

b) Scrieți valorile pe care le poate avea x , astfel încât în urma apelului $f(x)$, să se afișeze pe ecran exact 10 numere.

(6p.)

```
void f(int x)
{
    if (x<=10)
        cout<<0<<" "; | printf("%d ",0);
    else
    { f(x-2);
      cout<<x<<" "; | printf("%d ",x);
    }
}
```

V93

2. Subprogramul f este definit alăturat.

Ce se afișează ca urmare a apelului de mai jos?

 $f(1,4)$;

(6p.)

```
void f (int x,int y)
{ if (x<y){x=x+1;f(x,y);
  y=y-1;f(x,y);
}
else
  cout<<x<<y; | printf("%d%d",x,y);
}
```

V94

2. Funcția f este astfel definită încât $f(1)=8$, iar $f(n+1)=2*f(n)-4$ (n natural, $n>1$).
- a) Ce valoare are $f(5)$? (3p.)
- b) Care este cea mai mare valoare pe care o poate lua x astfel încât $f(x) < 1000$? (3p.)

V95

2. Funcția f are definiția alăturată. Scrieți cinci valori de apel pe care le poate avea n astfel încât, pentru cele 5 apeluri corespunzătoare acestor valori, să se obțină 5 valori ale funcției, distincte două câte două. (6p.)

```
int f(int n)
{
    if (n<=9) return 0;
    if (n%5==0) return 0;
    return 1+f(n-3);
}
```

V96

2. Funcția f are definiția alăturată. Scrieți 4 valori de apel pe care le poate avea n astfel încât, pentru cele 4 apeluri, corespunzătoare acestor valori, să se obțină 4 valori, distincte două câte două. (6p.)

```
int f(int n)
{
    if (n<=9) return 0;
    if (n%4==0) return 0;
    return 1+f(n-3);
}
```

V97

1. Subprogramul recursiv alăturat este definit incomplet: Care dintre următoarele expresii poate înlocui punctele de suspensie astfel încât, în urma apelului, subprogramul f să returneze suma primelor două cifre ale numărului primit prin intermediul parametrului x .
Exemplu: în urma apelului $f(2318)$ valoarea returnată este 5. (4p.)
- a. $x \leq 100$ b. $x \leq 99$ c. $x == 99$ d. $x != 0$

```
int f(int x){
    if (...)
        return x%10 + x/10;
    else
        return f(x/10);
}
```

V98

1. Se consideră subprogramul recursiv alăturat, definit incomplet. Cu ce valoare trebuie înlocuite punctele de suspensie, pentru ca funcția să returneze cifra minimă a numărului natural nenul transmis prin intermediul parametrului x ? (4p.)
- a. -1 b. 1 c. 9 d. 0

```
int Min(int x){
    int c;
    if (x==0) return ...;
    else {
        c=Min(x/10);
        if (c < x%10) return c;
        else return x%10;
    }
}
```

V99

1. Se consideră subprogramul recursiv alăturat, S , definit incomplet. Cu ce expresie pot fi înlocuite punctele de suspensie astfel încât, în urma apelului $S(2)$, să se afișeze 3 caractere $*$? (4p.)
- a. $x > 1$ b. $x > 2$ c. $x \geq 3$ d. $x > 0$

```
void S(int x)
{ cout<<'*';
  if (...) {
      cout<<'*';
      S(x-1);
  }
}
```

2. Ce afișează subprogramul **F**, descris alăturat, la apelul **F(5)**? (6p.)

```
void F(int x)
{
    cout<<x; | printf("%d",x);
    if(x>=3)
        F(x-2);
    cout<<x-1; | printf("%d",x-1);
}
```